Chapter 7
Three-Dimensional Geometric Transformations

➢ Main Content

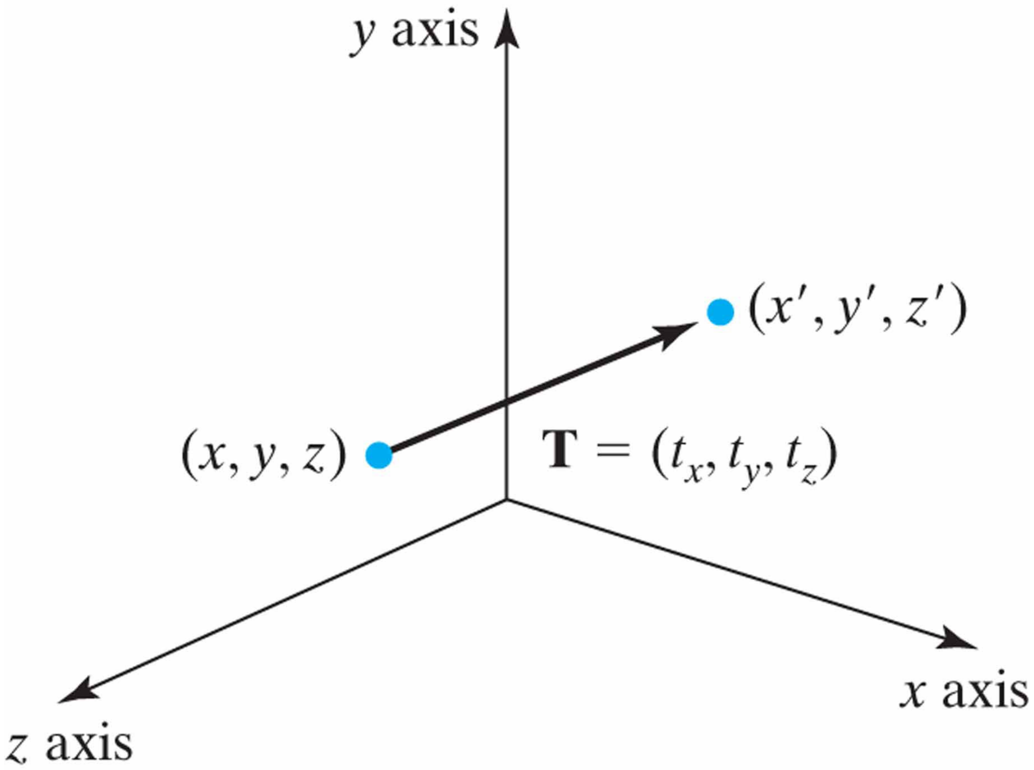
- 7.1 Three-Dimensional Translation
- 7.2 Three-Dimensional Rotation
- 7.3 Three-Dimensional Scaling
- 7.4 Composite Three-Dimensional Transformations
- 7.5 Other Three-Dimension Transformations
- 7.6 Transformations between Three-Dimensional Coordinate Systems
- 7.7 Affine Transformations

7.1 Three-Dimensional Translation

- A position $\mathbf{P} = (x, y, z)$ in three-dimensional space is translated to a location $\mathbf{P}' = (x', y', z')$ by adding translation distances t_x, t_y, and t_z to the Cartesian coordinates of $\mathbf{P}$:

$$x' = x + t_x, \quad y' = y + t_y, \quad z' = z + t_z$$

Moving a coordinate position with translation vector $T = (t_x, t_y, t_z)$



Three-Dimensional Translation

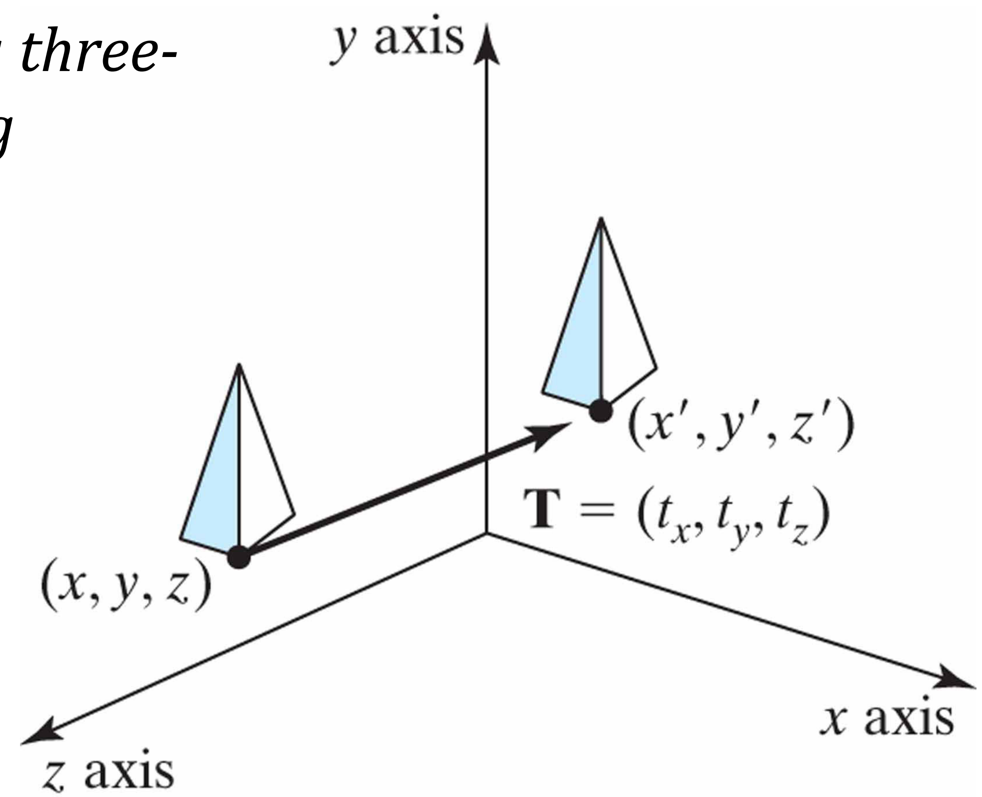
- We can express these three-dimensional translation operations in matrix form.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

or

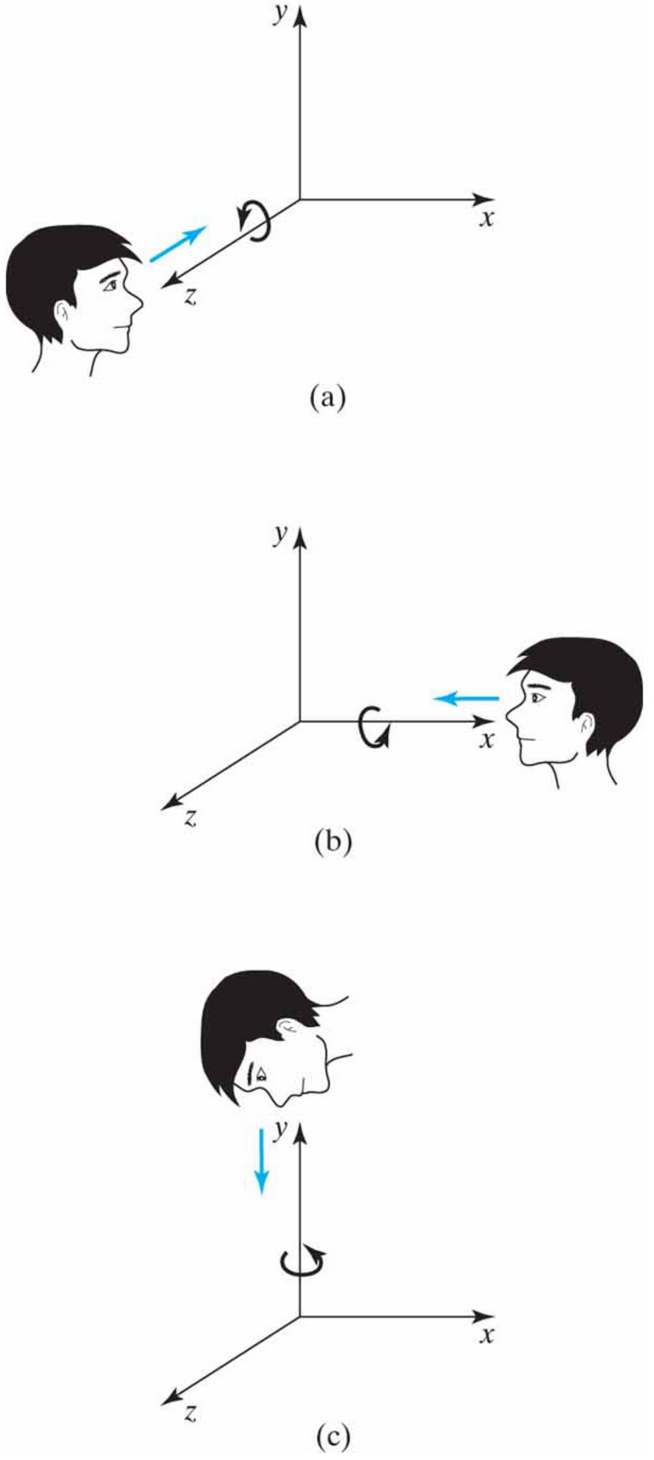
$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{P}$$

*Shifting the position of a three-dimensional object using translation vector **T**.*



7.2 Three-Dimensional Rotation

- By convention, positive rotation angles produce counterclockwise rotations about a coordinate axis, assuming that we are looking in the negative direction along that coordinate axis.



- **Three-dimensional Coordinate-Axis Rotations**
  - The two-dimensional z-axis rotation equations are easily extended to three dimensions, as follows:

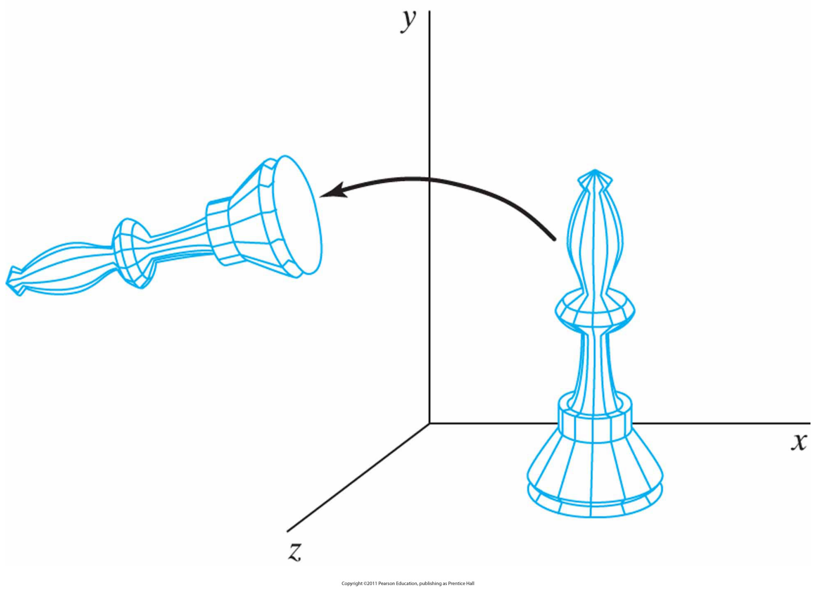
$$x' = x\cos\theta - y\sin\theta \quad y' = x\sin\theta + y\cos\theta \quad z' = z$$

In homogeneous-coordinate for, the three-dimensional z-axis rotation equations are

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

which can write more compactly as:

$$\mathbf{P}' = \mathbf{R}_{\boldsymbol{z}}(\boldsymbol{\theta}) \cdot \mathbf{P}$$



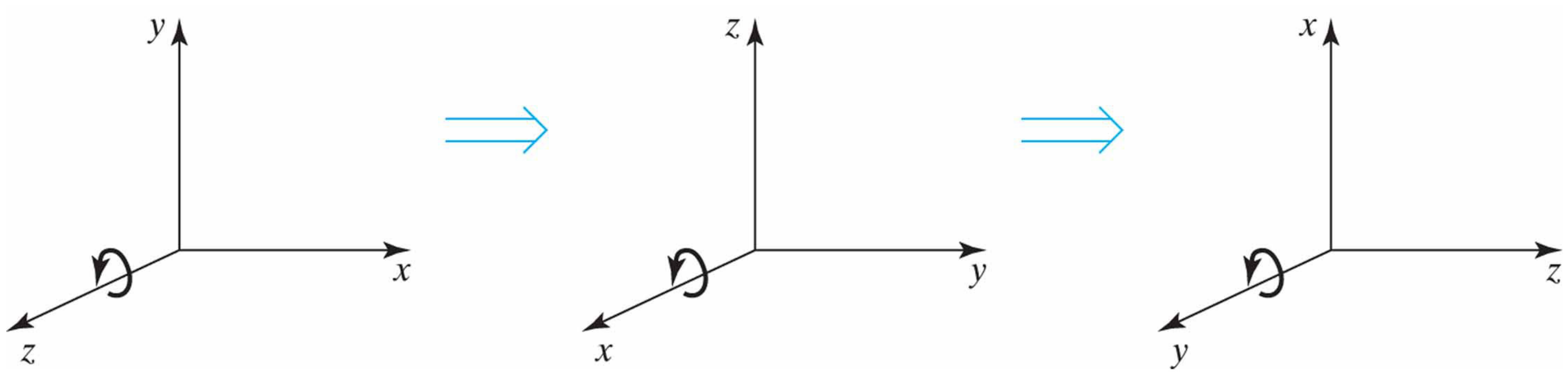
- **Three-dimensional Coordinate-Axis Rotations**
  - Transformation equations for rotations about the other two coordinate axes can be obtained with a cyclic permutation of the coordinate parameters x, y. and z :

$$x \longrightarrow y \longrightarrow z \longrightarrow x$$

  - Thus, we get the equations for an other axes rotation:

$$y' = y\cos\theta - z\sin\theta \qquad z' = y\sin\theta + z\cos\theta \qquad x' = x$$

$$z' = z\cos\theta - x\sin\theta \qquad x' = z\sin\theta + x\cos\theta \qquad y' = y$$



- **General Three-dimensional Rotations**

  - A rotation matrix for any axis that does not coincide with a coordinate axis can be set up as a composite transformation involving combinations of translations and the coordinate-axis rotations.

  - **Case 1**: Parallel

  - **Case 2**: General

• General Three-dimensional Rotations

Case 1: Parallel

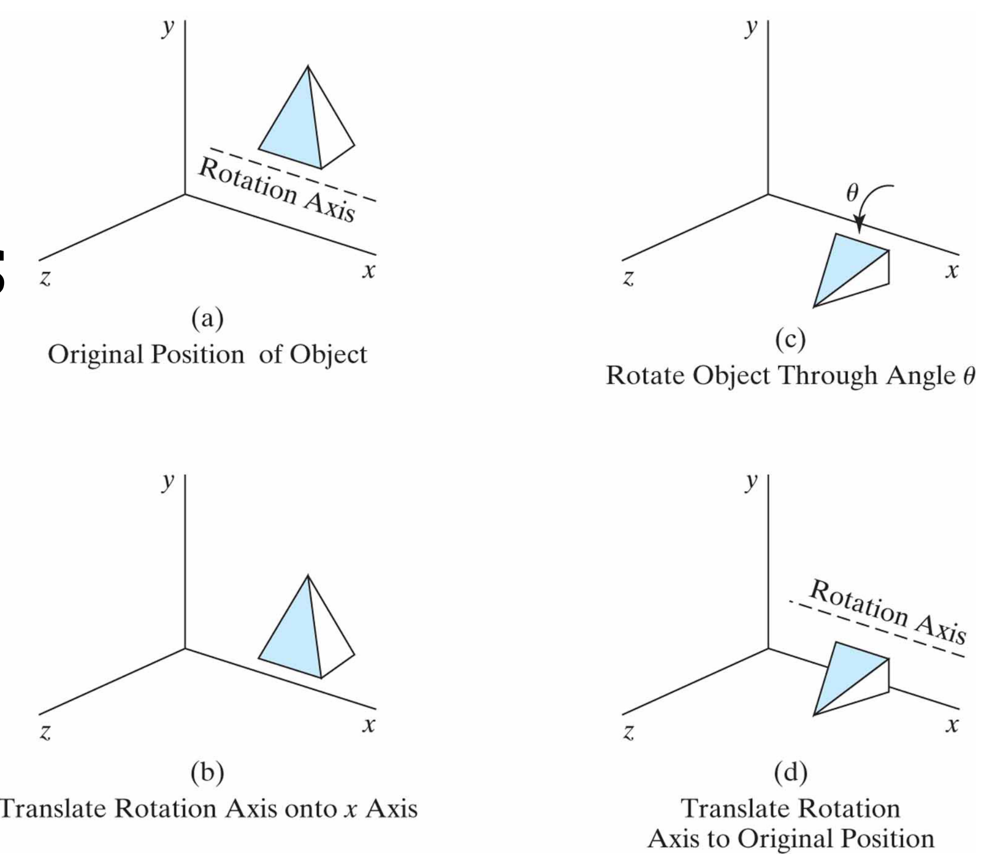
- In the special case where an object is to be rotated about an axis that is parallel to one of the coordinate axes:
    1. *Translate the object so that the rotation axis coincides with the parallel coordinate axis.*
    2. *Perform the specified rotation about that axis.*
    3. *Translate the object so that the rotation axis is moved back to its original position.*

- A coordinate position **P** is transformed with the sequence as

$$\mathbf{P}' = \mathbf{T}^{-1}\mathbf{R}_x(\theta) \cdot \mathbf{T} \cdot \mathbf{P}$$

The composite rotation matrix:

$$\mathbf{R}(\theta) = \mathbf{T}^{-1}\mathbf{R}_x(\theta) \cdot \mathbf{T}$$



(a) Original Position of Object

(b) Translate Rotation Axis onto *x* Axis

(c) Rotate Object Through Angle θ

(d) Translate Rotation Axis to Original Position

- **General Three-dimensional Rotations**

Case 2: General

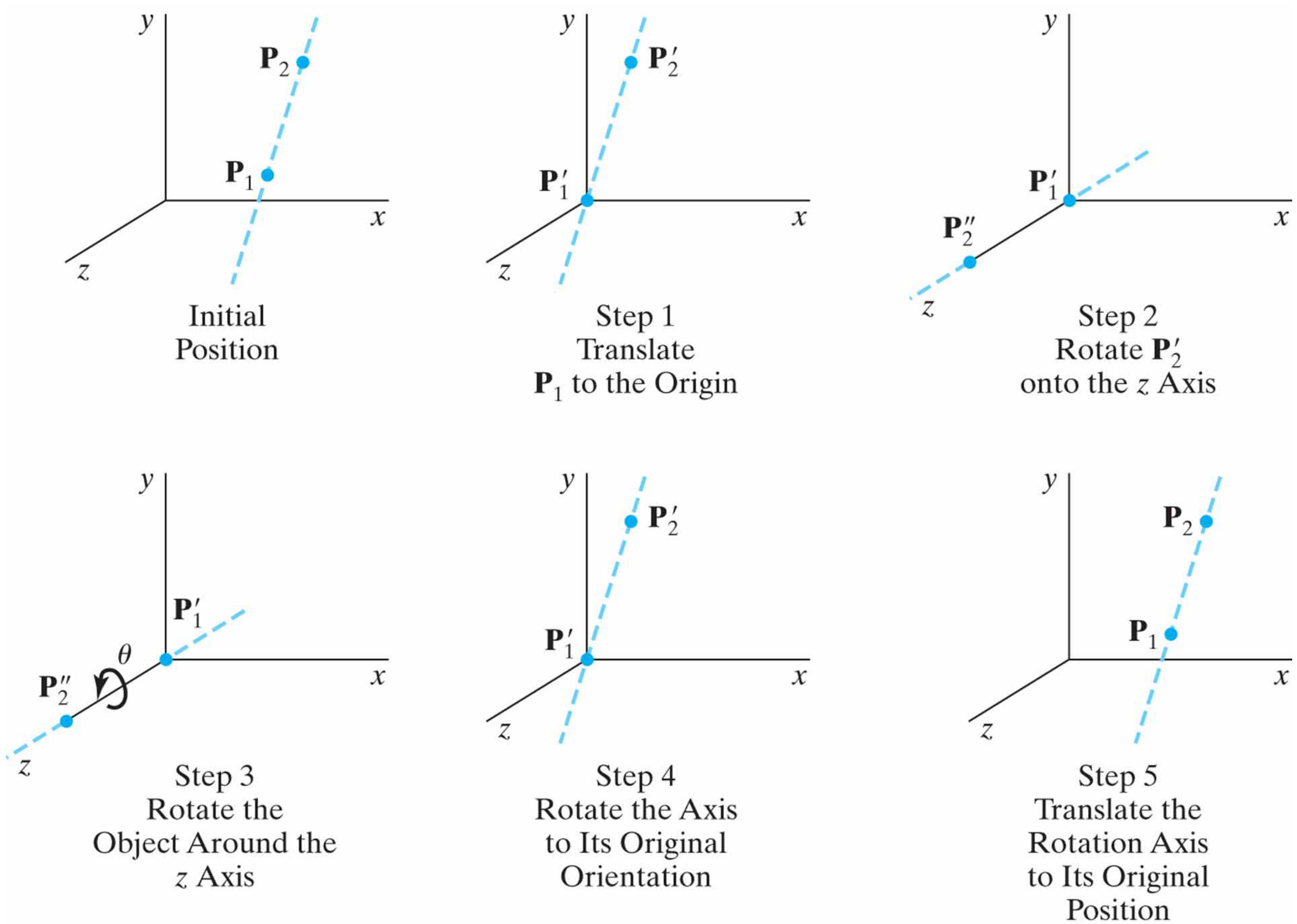
- Given the specifications for the rotation axis and the rotation angle, we can accomplish the required rotation in five steps

1. *Translate the object so that the rotation axis passes through the coordinate origin.*
2. *Rotate the object so that the axis of rotation coincides with one of the coordinate axes*
3. *Perform the specified rotation about the selected coordinate axis.*
4. *Apply inverse rotations to bring the rotation axis back to its original orientation*
5. *Apply the inverse translation to bring the rotation axis back to its original spatial position.*

- **General Three-dimensional Rotations**

Case 2: General

➢ We can transform the rotation axis onto any one of the three coordinate axes. We next consider a transformation sequence using the z-axis rotation matrix.



- **General Three-dimensional Rotations**

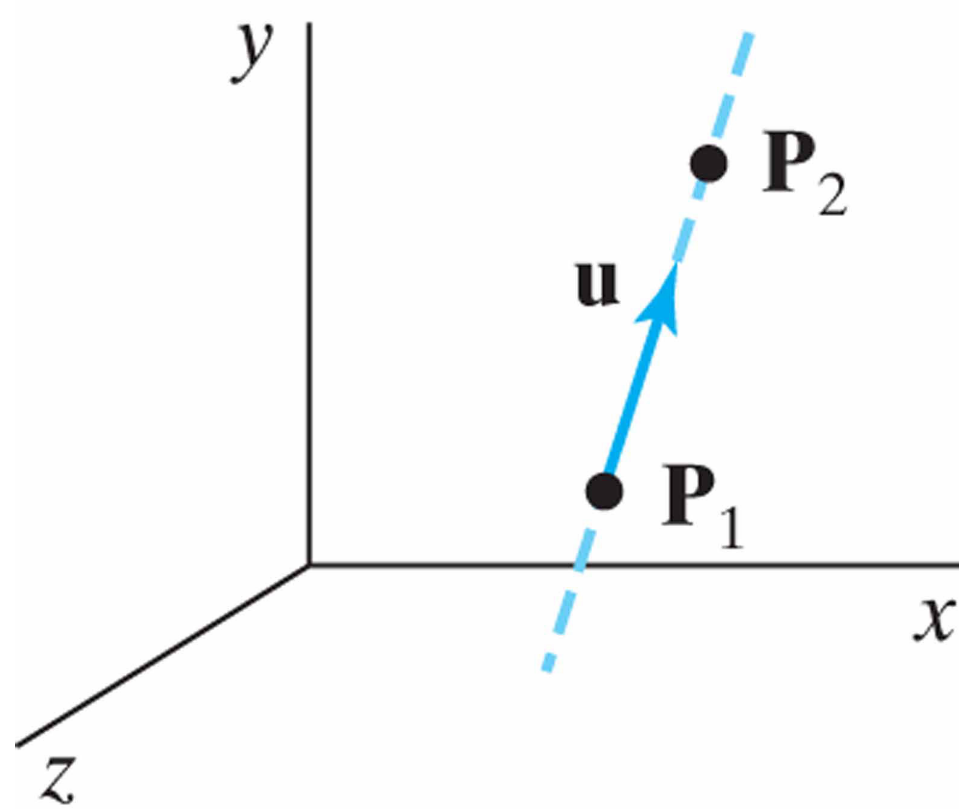
Case 2: General

- We assume that the rotation axis is defined by two points and the components of the rotation-axis vector are then computed as:

$$\mathbf{V} = \mathbf{P_2} - \mathbf{P_1} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

- The unit rotation-axis vector **u** is

$$\mathbf{u} = \frac{\mathbf{V}}{|\mathbf{V}|} = (a, b, c)$$



where the components a, b, and c are the direction cosines for the rotation axis

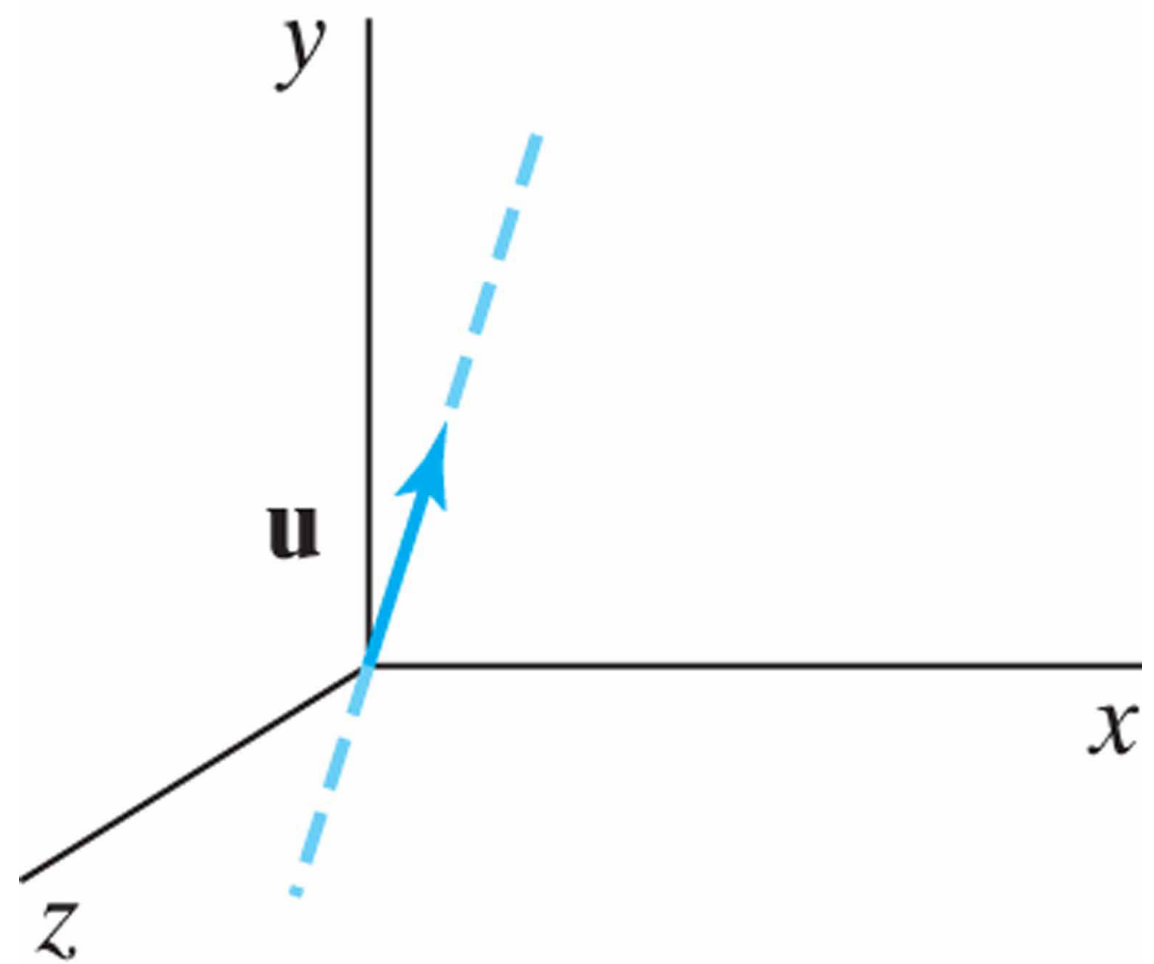
$$a = \frac{x_2 - x_1}{|\mathbf{V}|} \quad b = \frac{y_2 - y_1}{|\mathbf{V}|} \quad c = \frac{z_2 - z_1}{|\mathbf{V}|}$$

- **General Three-dimensional Rotations**

First Step

➢ Set up the translation matrix that repositions the rotation axis so that it passes through the coordinate origin.

Translation of the rotation axis to the coordinate origin



The translation matrix is

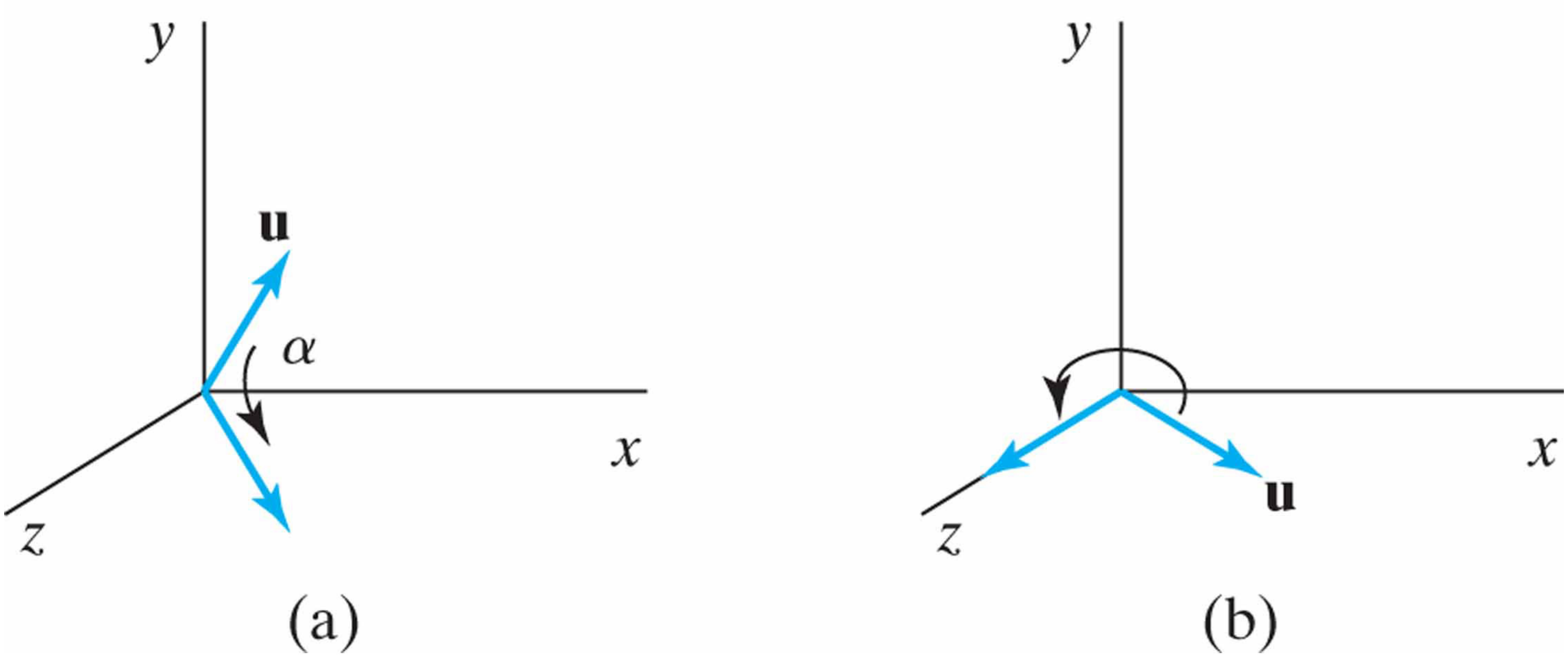
$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **General Three-dimensional Rotations**

Second Step

- Formulate the transformation that will put the rotation axis onto the z axis.

Unit vector u is rotated about the x axis to bring it into the xz plane (a), then it is rotated around the y axis to align it with the z axis (b).



- **General Three-dimensional Rotations**

Second Step-A

- We establish the transformation matrix for rotation around the x axis by determining the values for the sine and cosine of the rotation angle necessary to get **u** into the xz plane.

Rotation of u around the x axis into the xz plane is accomplished by rotating $\mathbf{u}'$ *(the projection of u in the yz plane) through angle α onto the z axis.*

$$\cos \alpha = \frac{\mathbf{u}' \cdot \mathbf{u}_z}{|\mathbf{u}' \cdot \mathbf{u}_z|} = \frac{c}{d}$$

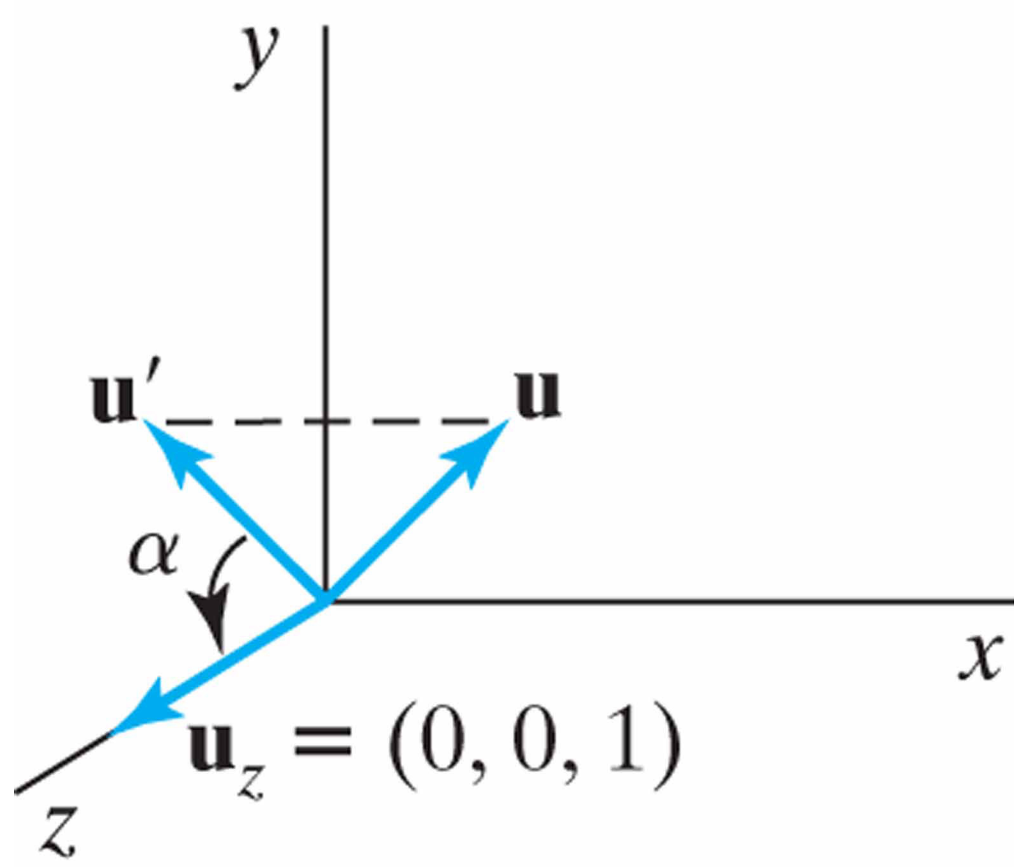
where d is the magnitude of $\mathbf{u}'$

$$d = \sqrt{b^2 + c^2}$$

Similarly, we can determine the sine of α

$$\mathbf{u}' \times \mathbf{u}_z = \mathbf{u}_x |\mathbf{u}'||\mathbf{u}_z| \sin \alpha = \mathbf{u}_x \cdot b$$

$$\sin \alpha = \frac{b}{d}$$

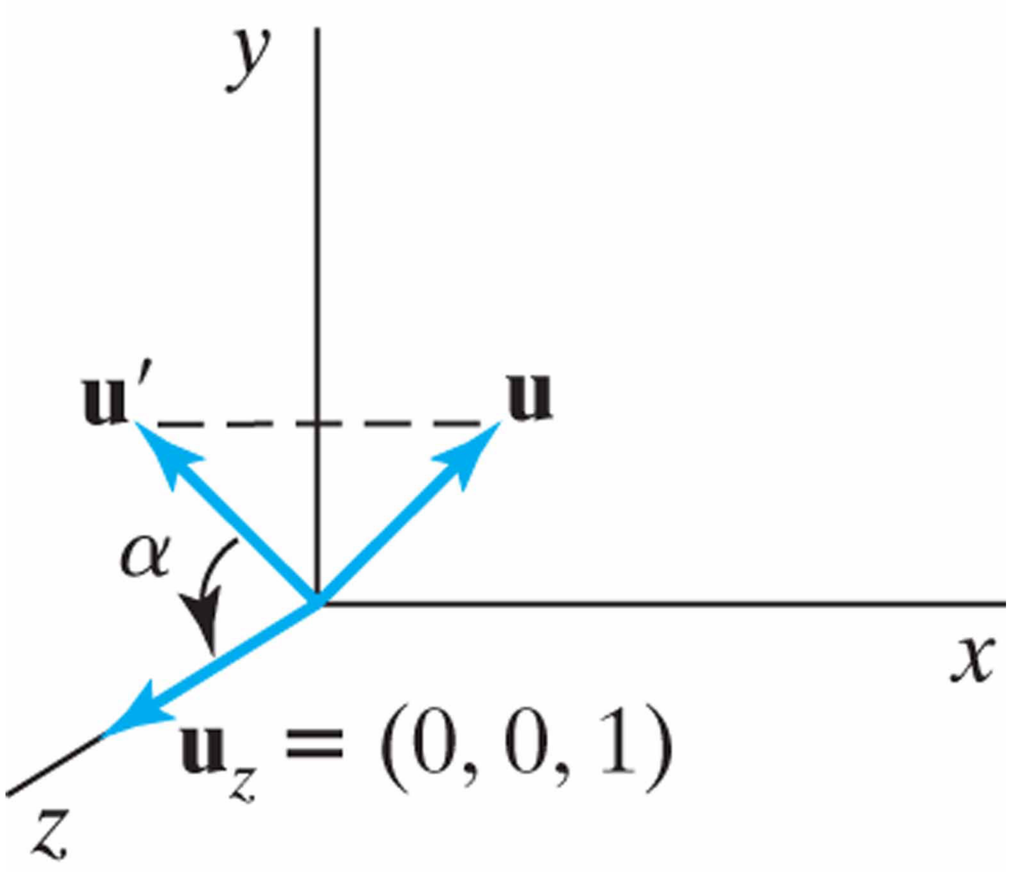


- # General Three-dimensional Rotations

Second Step-A

➢ Now that we have determined the values for cos α and sin α in terms of the components of vector **u**, we can set up the matrix elements for rotation:

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{b}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



• General Three-dimensional Rotations

Second Step-B

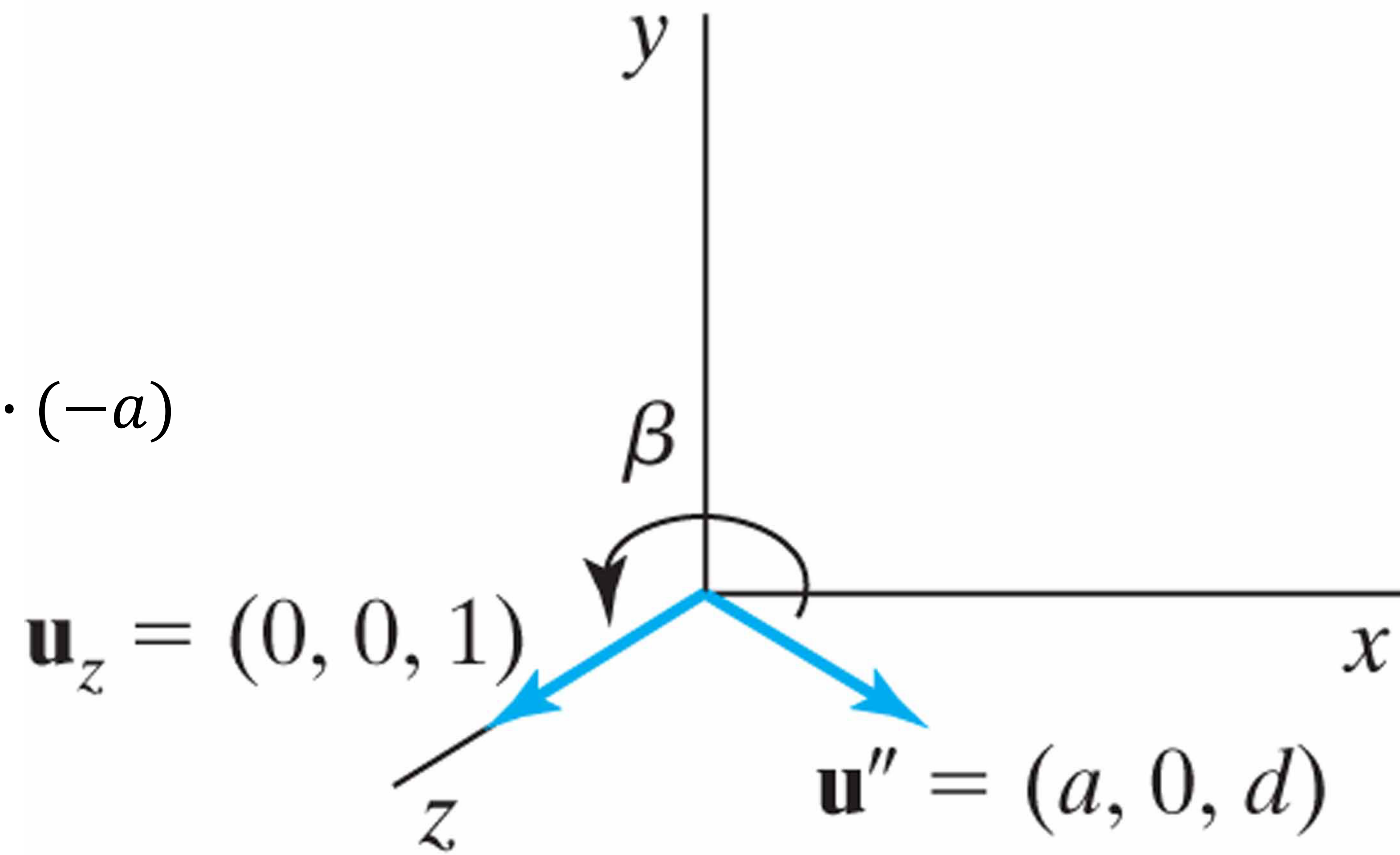
- Determine the matrix that will swing the unit vector in the xz plane counterclockwise around the y axis onto the positive z axis.

*Rotation of unit vector **u**'' (vector u after rotation into the xz plane) about the y axis. Positive rotation angle β aligns **u**'' with vector $\mathbf{u}_z$.*

$$\cos\beta = \frac{\mathbf{u}'' \cdot \mathbf{u}_z}{|\mathbf{u}'' \cdot \mathbf{u}_z|} = d$$

$$\mathbf{u}'' \times \mathbf{u}_z = \mathbf{u}_y |\mathbf{u}''||\mathbf{u}_z| \sin\beta = \mathbf{u}_y \cdot (-a)$$

$$\sin\beta = -a$$

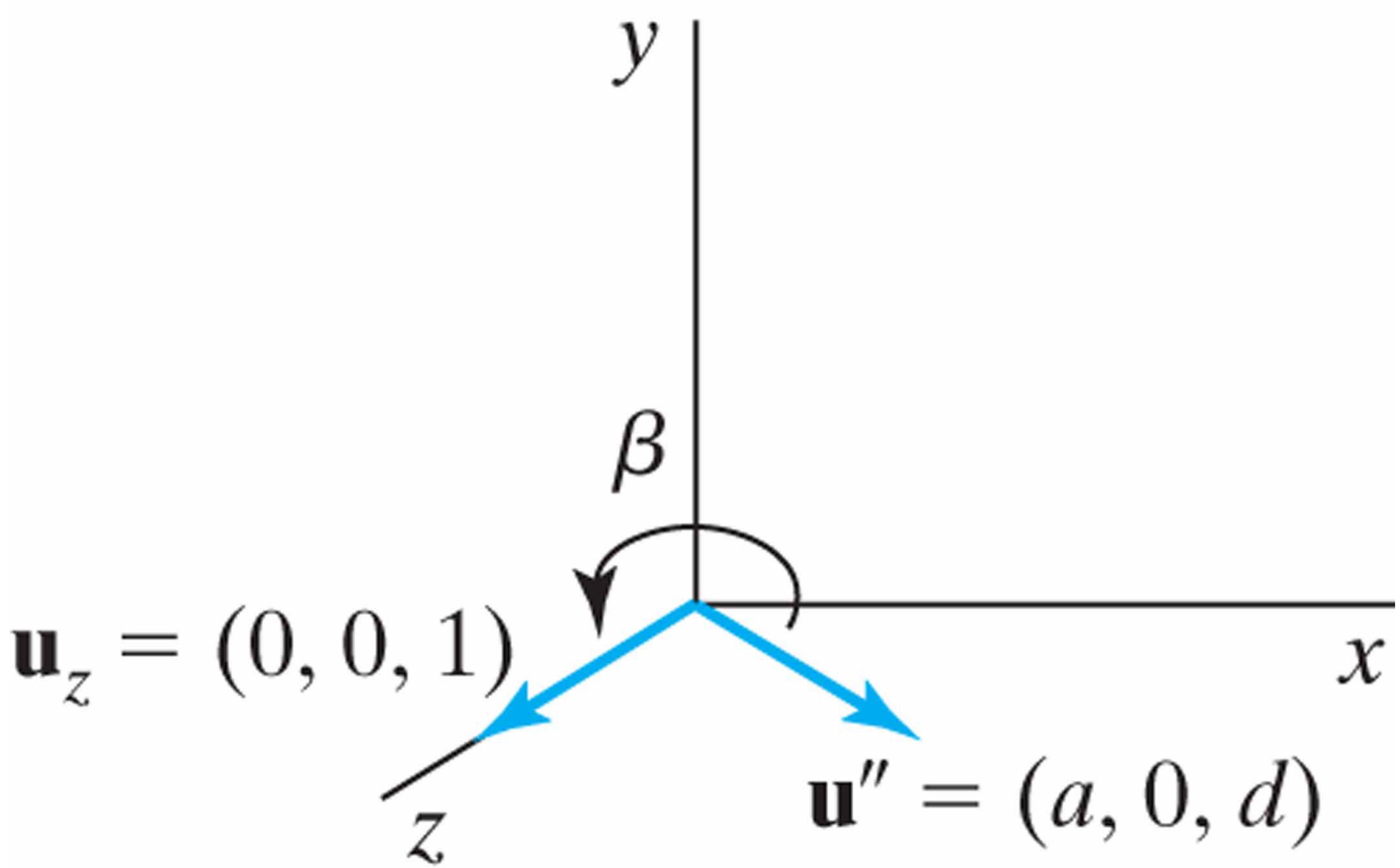


- **General Three-dimensional Rotations**

Second Step-B

➢ Therefore, the transformation matrix for rotation of $\mathbf{u}''$ about the y axis is

$$\mathbf{R}_{y}(\beta) = \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- **General Three-dimensional Rotations**

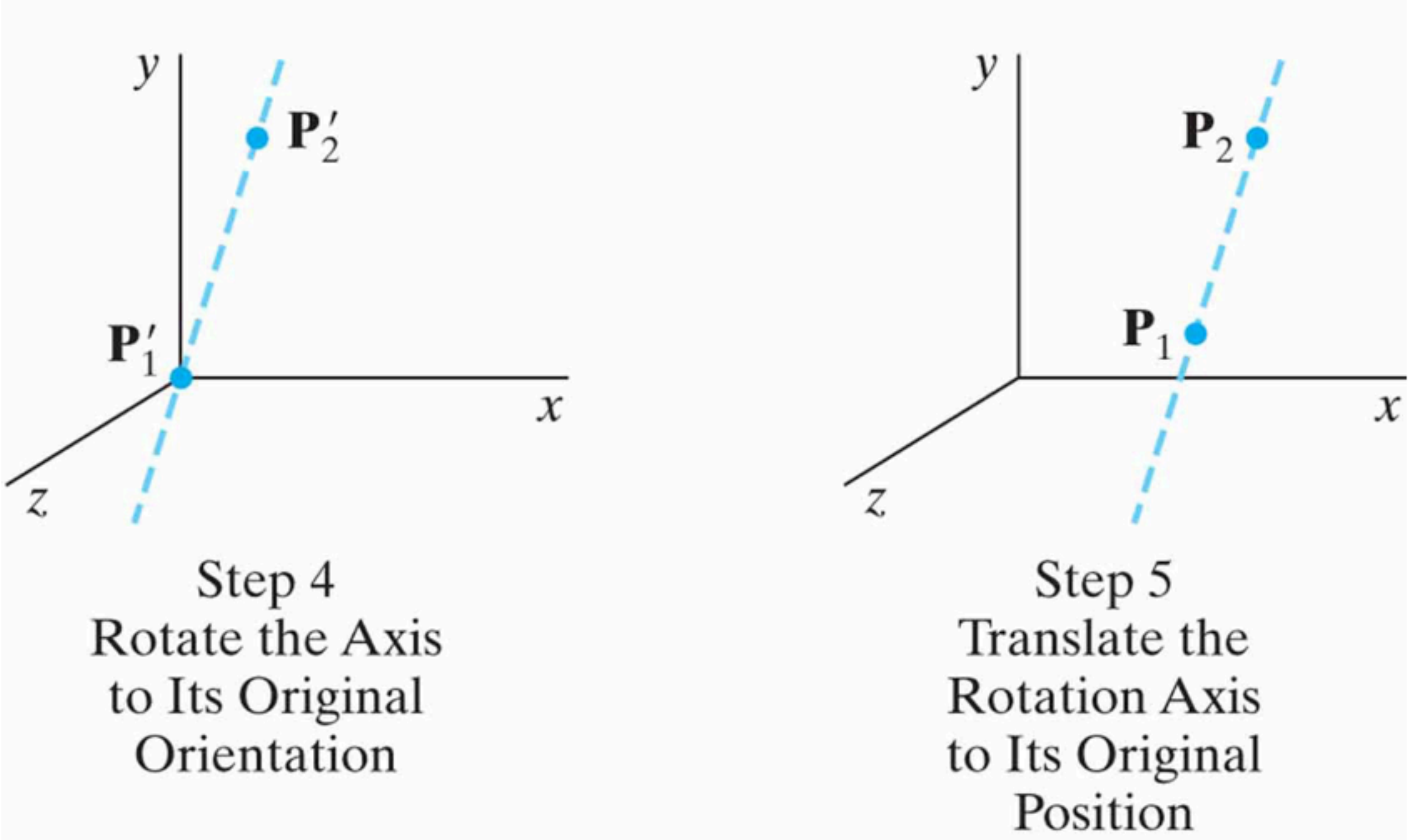
Third Step

- With *Second Step*, we have aligned the rotation axis with the positive z axis.
- The specified rotation angle θ can now be applied as a rotation about the z axis as follows:

$$\mathbf{R}_{\boldsymbol{z}}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **General Three-dimensional Rotations**

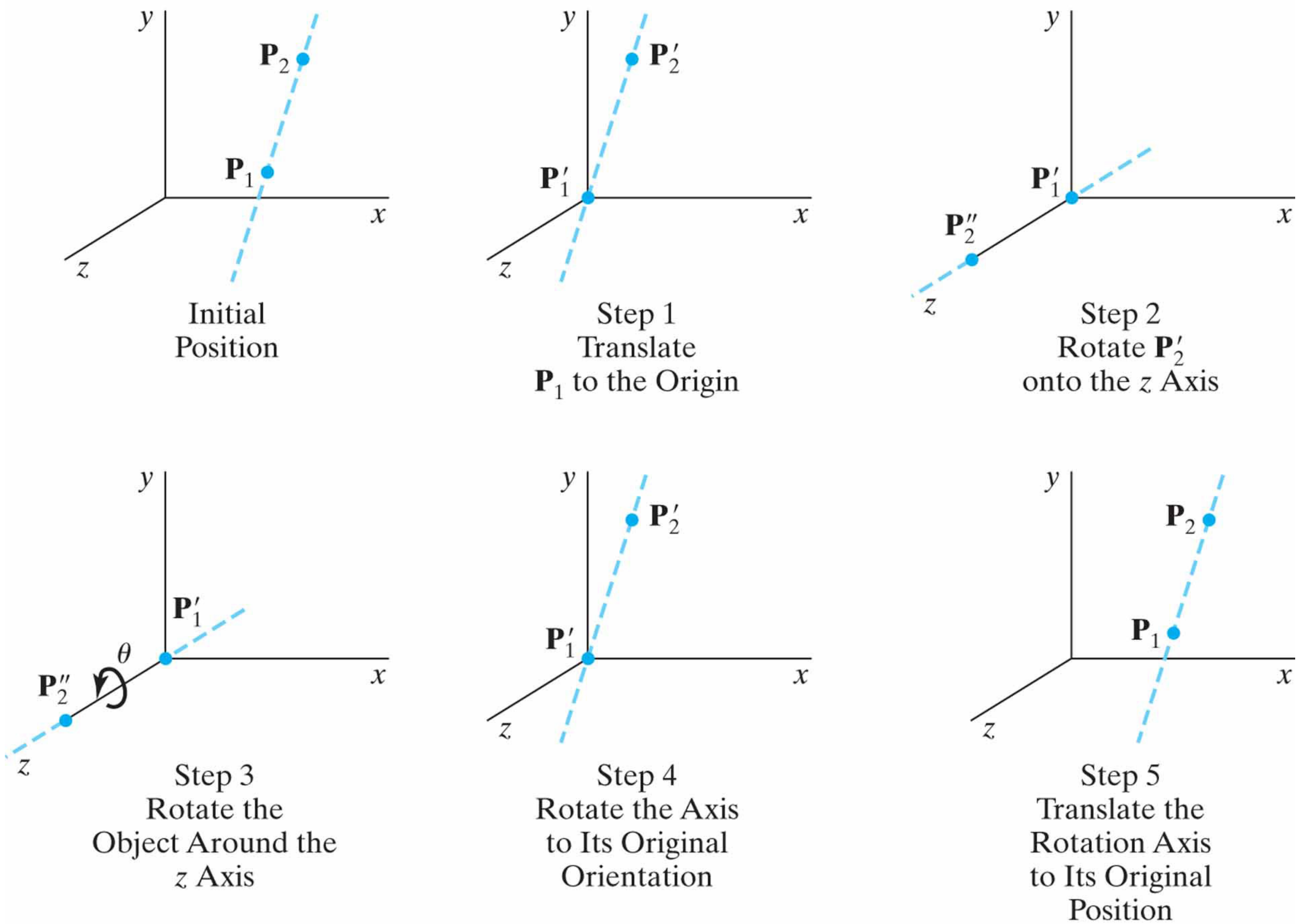
Inverse Step



- To complete the required rotation about the given axis, we need to transform the rotation axis back to its original position. This is done by applying the inverse of transformations: $\mathbf{T}^{-1}$, $\mathbf{R}_x^{-1}(\alpha)$ and $\mathbf{R}_y^{-1}(\beta)$

- **General Three-dimensional Rotations**

Five Steps Approach



$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R}_x^{-1}(\alpha) \cdot \mathbf{R}_y^{-1}(\beta) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\alpha) \cdot \mathbf{T}$$

- **General Three-dimensional Rotations**

Quicker Second Step

- A somewhat quicker method for obtaining the composite rotation matrix $\mathbf{R}_y(\beta)\cdot\mathbf{R}_x(\alpha)$ is to use the fact that the composite matrix for any sequence of three-dimensional rotations is of the form:

$$\mathbf{R}=\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The upper-left 3X3 submatrix of this matrix is orthogonal:

$$\mathbf{R}\cdot\begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ 1 \end{bmatrix}=\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{R}\cdot\begin{bmatrix} r_{21} \\ r_{22} \\ r_{23} \\ 1 \end{bmatrix}=\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{R}\cdot\begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \\ 1 \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

- **General Three-dimensional Rotations**

Quicker Second Step

➢ Assuming that the rotation axis is not parallel to any coordinate axis, we could form the following set of local unit vectors:

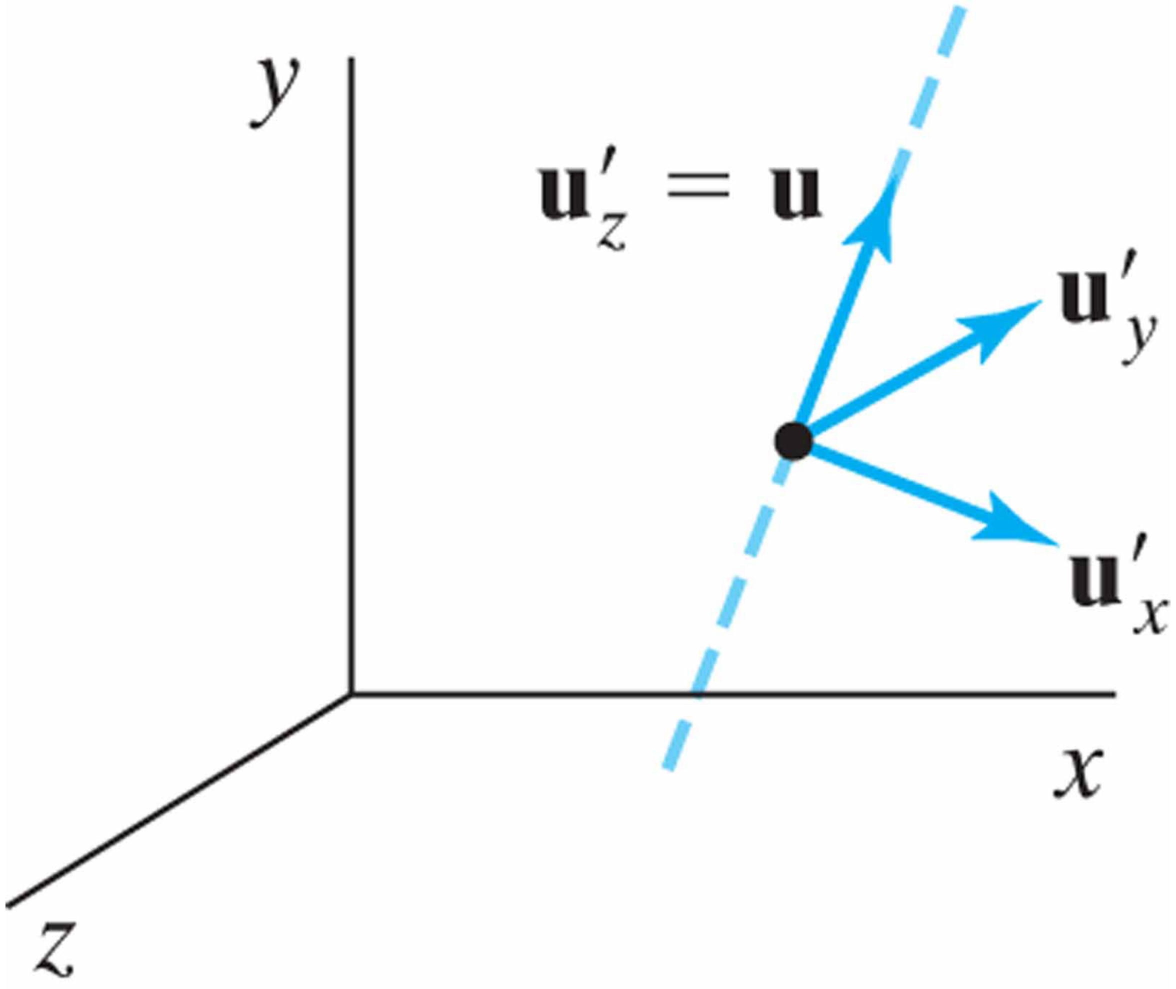
$$\mathbf{u}'_{\mathrm{z}} = \mathbf{u} = (u'_{z1}, u'_{z2}, u'_{z3})$$

$$\mathbf{u}'_{\mathrm{y}} = \frac{\mathbf{u} \times \mathbf{u}_x}{|\mathbf{u} \times \mathbf{u}_x|} = (u'_{y1}, u'_{y2}, u'_{y3})$$

$$\mathbf{u}'_x = \mathbf{u}'_{\mathrm{y}} \times \mathbf{u}'_{\mathrm{z}} = (u'_{x1}, u'_{x2}, u'_{x3})$$

Then the required composite matrix, which is equal to $\boldsymbol{R_y}(\beta) \cdot \boldsymbol{R_x}(\alpha)$ *is*

$$\mathbf{R} = \begin{bmatrix} u'_{x1} & u'_{x2} & u'_{x3} & 0 \\ u'_{y1} & u'_{y2} & u'_{y3} & 0 \\ u'_{z1} & u'_{z2} & u'_{z3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- **Quaternion Method for Three-dimensional Rotations**

  - A more efficient method for generating a rotation about an arbitrarily selected axis is to use a quaternion representation for the rotation transformation.

  - This is particular important in animations, which often require complicated motion sequences and motion interpolations between two given positions of an object.

- **Quaternion Method for Three-dimensional Rotations**

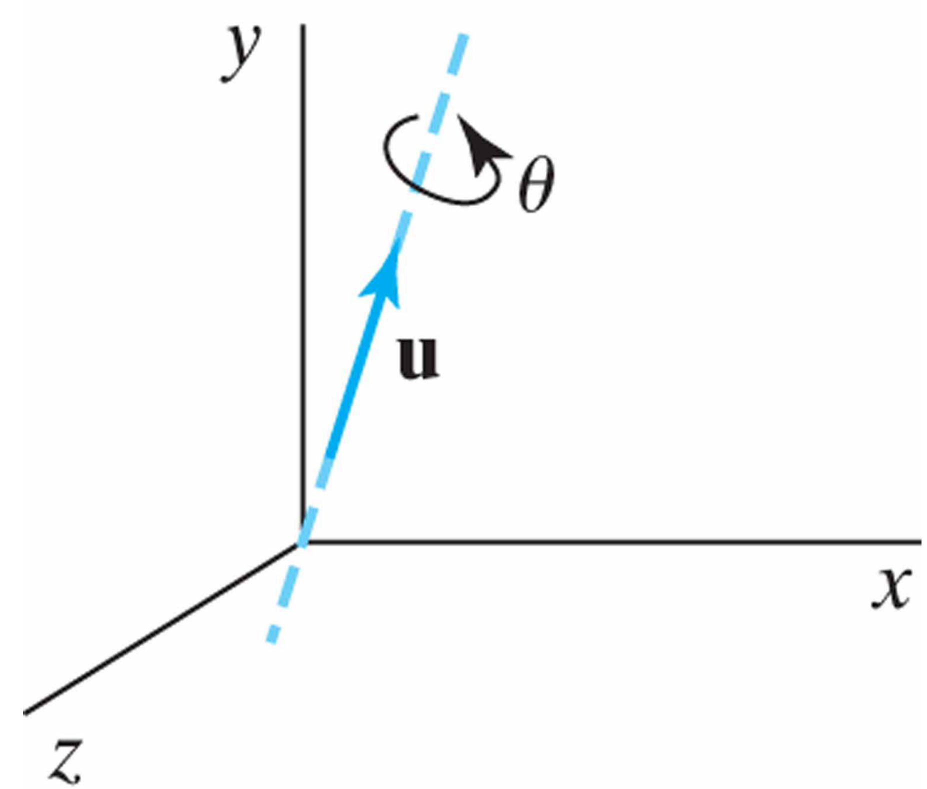
  - One way to character a quaternion is as an ordered pair, consisting of a scalar part and a vector part:

$$q = (s, \mathbf{v})$$

  - A rotation about any axis passing though the coordinate origin is accomplished by first setting:

$$s = \cos\frac{\theta}{2}$$

$$\mathbf{v} = \mathbf{u}\sin\frac{\theta}{2}$$



- **Quaternion Method for Three-dimensional Rotations**

  - Any point position **P** that is to be rotated can be represented in quaternion notation as:

$$\mathbf{P} = (0, \mathbf{p})$$

  - The rotation of the point is then carried out with the quaternion operation:

$$\mathbf{P}' = q\mathbf{P}q^{-1} = (0, \mathbf{p}')$$

where $q^{-1} = (s, -\boldsymbol{v})$ *is the inverse of the unit.*

  - The second term in this ordered pair is the rotated point position **p'** as

$$\mathbf{P}' = s^2\mathbf{p} + \mathbf{v}(\mathbf{p} \cdot \mathbf{v}) + 2s(\mathbf{v}\times\mathbf{p}) + \mathbf{v}\times(\mathbf{v}\times\mathbf{p})$$

- **Quaternion Method for Three-dimensional Rotations**
  - Designating the components of the vector part of q as $\mathbf{v} = (a, b, c)$, we obtain the elements for the composite rotation matrix $\mathbf{R}_x^{-1}(\alpha) \cdot \mathbf{R}_y^{-1}(\beta) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{R}_{\boldsymbol{y}}(\beta) \cdot \mathbf{R}_{\boldsymbol{x}}(\alpha)$ in a 3X3 form as:

$$\mathbf{M}_{\boldsymbol{R}}(\theta) = \begin{bmatrix} 1 - 2b^2 - 2c^2 & 2ab - 2sc & 2ac + 2sb \\ 2ab + 2sc & 1 - 2a^2 - 2c^2 & 2bc - 2sa \\ 2ac - 2sb & 2bc + 2sa & 1 - 2a^2 - 2b^2 \end{bmatrix}$$

  - Using the following trigonometric identities to simplify the terms:

$$\cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2} = 1 - 2\sin^2\frac{\theta}{2} = \cos\theta$$

$$2\cos\frac{\theta}{2}\sin\frac{\theta}{2} = \sin\theta$$

- **Quaternion Method for Three-dimensional Rotations**
  - Thus, we can rewrite the matrix as:

$$\mathbf{M}_{\boldsymbol{R}}(\theta) =$$
$$\begin{bmatrix} u_x^2(1-\cos\theta)+\cos\theta & u_xu_y(1-\cos\theta)-u_z\sin\theta & u_xu_z(1-\cos\theta)+u_y\sin\theta \\ u_yu_x(1-\cos\theta)+u_z\sin\theta & u_y^2(1-\cos\theta)+\cos\theta & u_yu_z(1-\cos\theta)-u_x\sin\theta \\ u_zu_x(1-\cos\theta)-u_y\sin\theta & u_zu_y(1-\cos\theta)+u_x\sin\theta & u_z^2(1-\cos\theta)+\cos\theta \end{bmatrix}$$

where u_x, u_x, and u_x are the components of the unit axis vector ***u.***

  - Including the translations that move the rotation axis to the coordinate axis and return it to its original position, the complete quaternion rotation expression is:

$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{M}_R \cdot \mathbf{T}$$

7.3 Three-Dimensional Scaling

- The matrix expression for the three-dimensional scaling transformation of a position **P**= (x, y, z) relative to the coordinate origin is a simple extension of two-dimensional scaling.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Explicit expression for the scaling transformation relative to the origin are:

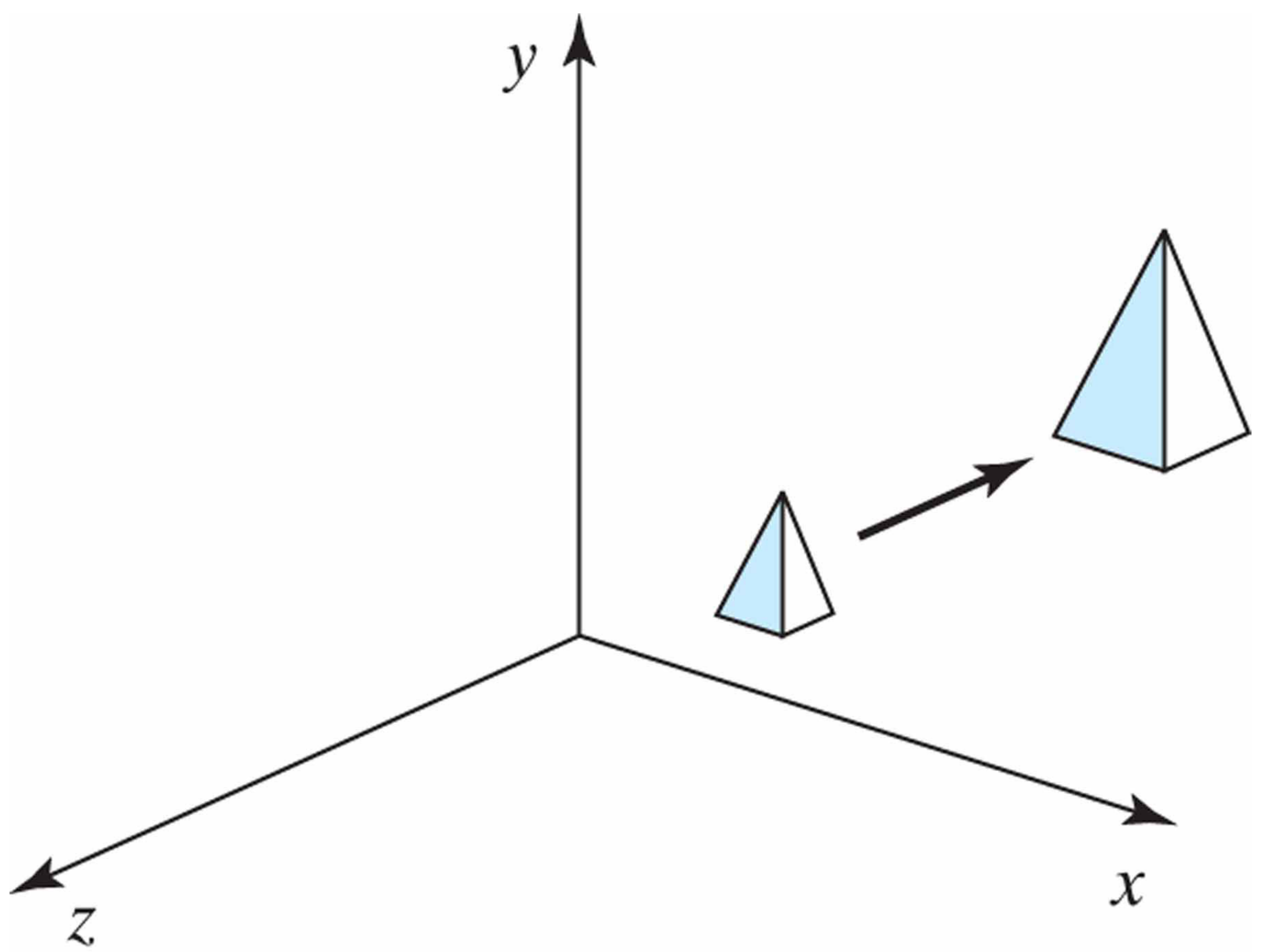
$$x' = x \cdot s_x \qquad y' = y \cdot s_y \qquad z' = z \cdot s_z$$

- **Three-dimensional Scaling**

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

  - ➢ Scaling an object changes the position of the object relative to the coordinate origin.

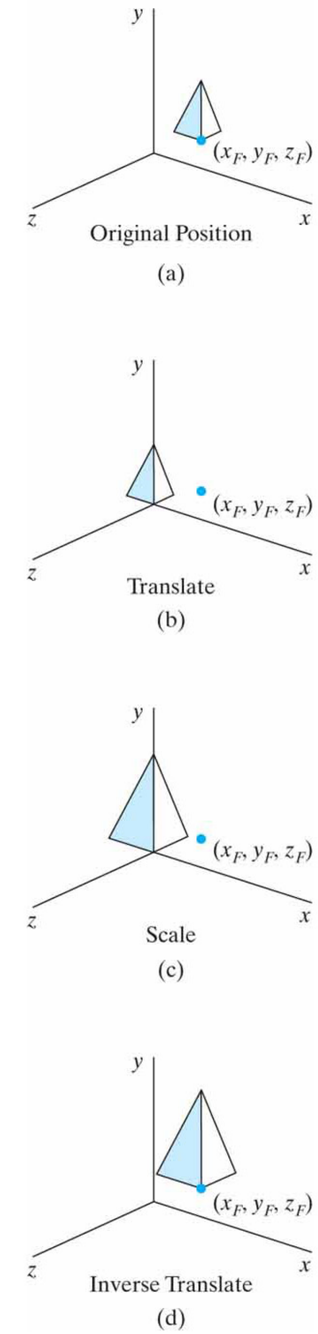
Doubling the size of an object with transformation 9-41 also moves the object farther from the origin



- **Three-dimensional Scaling**
  - We can always construct a scaling transformation with respect to any selected fixed position (x_f, y_f, z_f):

$$\mathbf{T}(x_f, y_f, z_f) \cdot \mathbf{S}(x_f, y_f, z_f) \cdot \mathbf{T}(-x_f, -y_f, -z_f)$$

$$= \begin{bmatrix} s_x & 0 & 0 & (1 - s_x)x_f \\ 0 & s_y & 0 & (1 - s_y)y_f \\ 0 & 0 & s_z & (1 - s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



7.4 Composite Three-Dimensional Transformation

- We can implement a transformation sequence by concatenating the individual matrices from right to left or from left to right, depending on the order in which the matrix representations are specified.

- The rightmost term in a matrix product is always the first transformation to be applied to an object and the leftmost term is always the last transformation.

7.5 Other Three-Dimensional Transformations

- Three-Dimensional Reflections

- Three-Dimensional Shears

- **Three-dimensional Reflections**

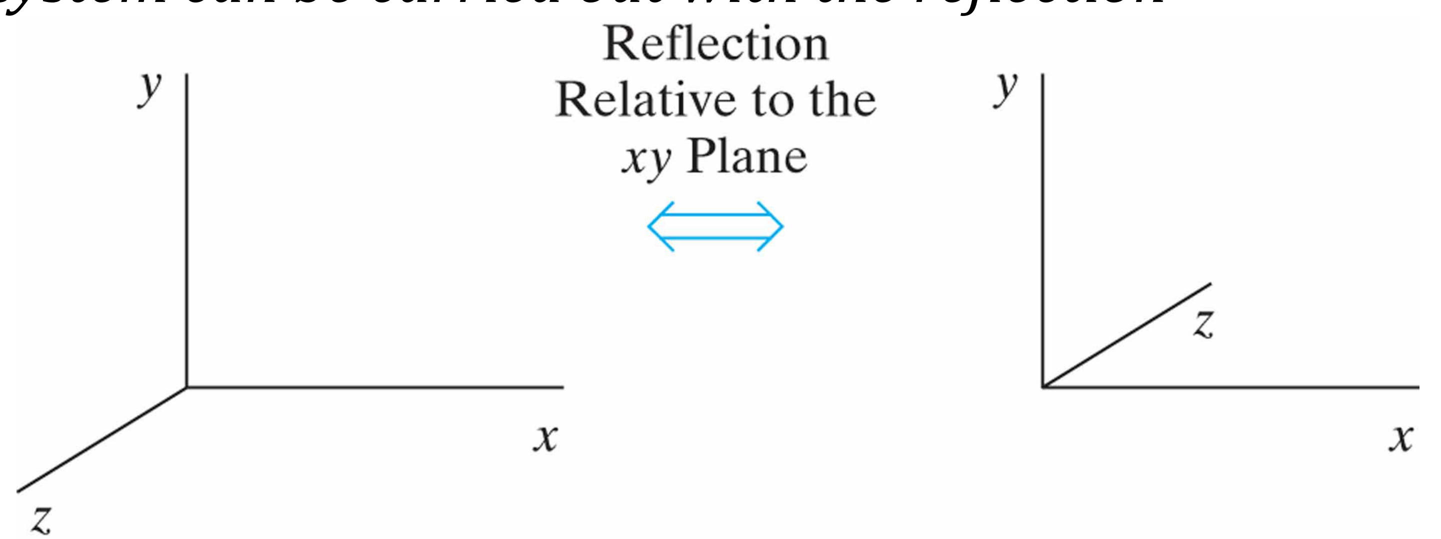
  - A reflection in a three-dimensional space can be performed relative to a selected *reflection axis* or with respect to a *reflection plane*.

  - Reflections relative to a given axis are equivalent to 180^0 rotations about that axis.

  - Reflections with respect to a plane are similar

- **Three-dimensional Reflections**
  - ➢ When the reflection plane is a coordinate plane (xy, xz, or yz), we can think of the transformation as a 180^0 rotation in four-dimensional space with a conversion between left-handed frame and a right-handed frame.

Conversion of coordinate specifications between a right-handed and a left-handed system can be carried out with the reflection transformation



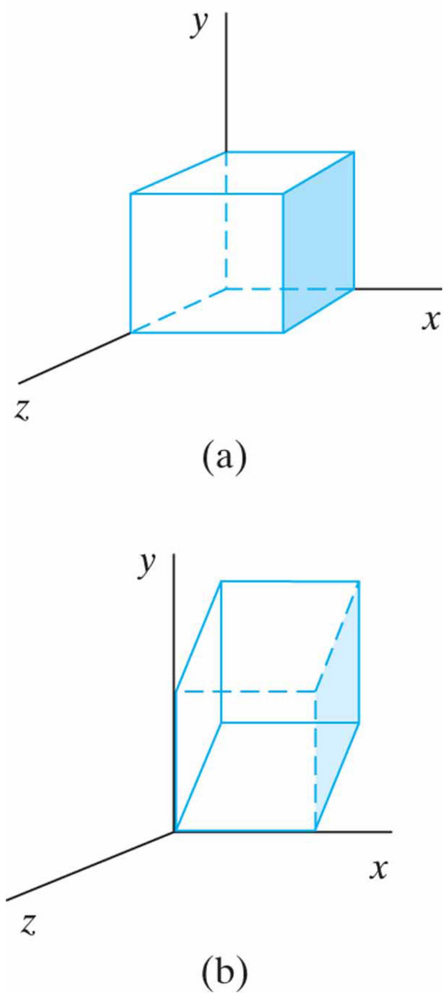
$$M_{zreflect} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Three-dimensional Shears**

➢ Can be used to modify object shapes, also applied in three-dimensional viewing transformations for perspective projections.

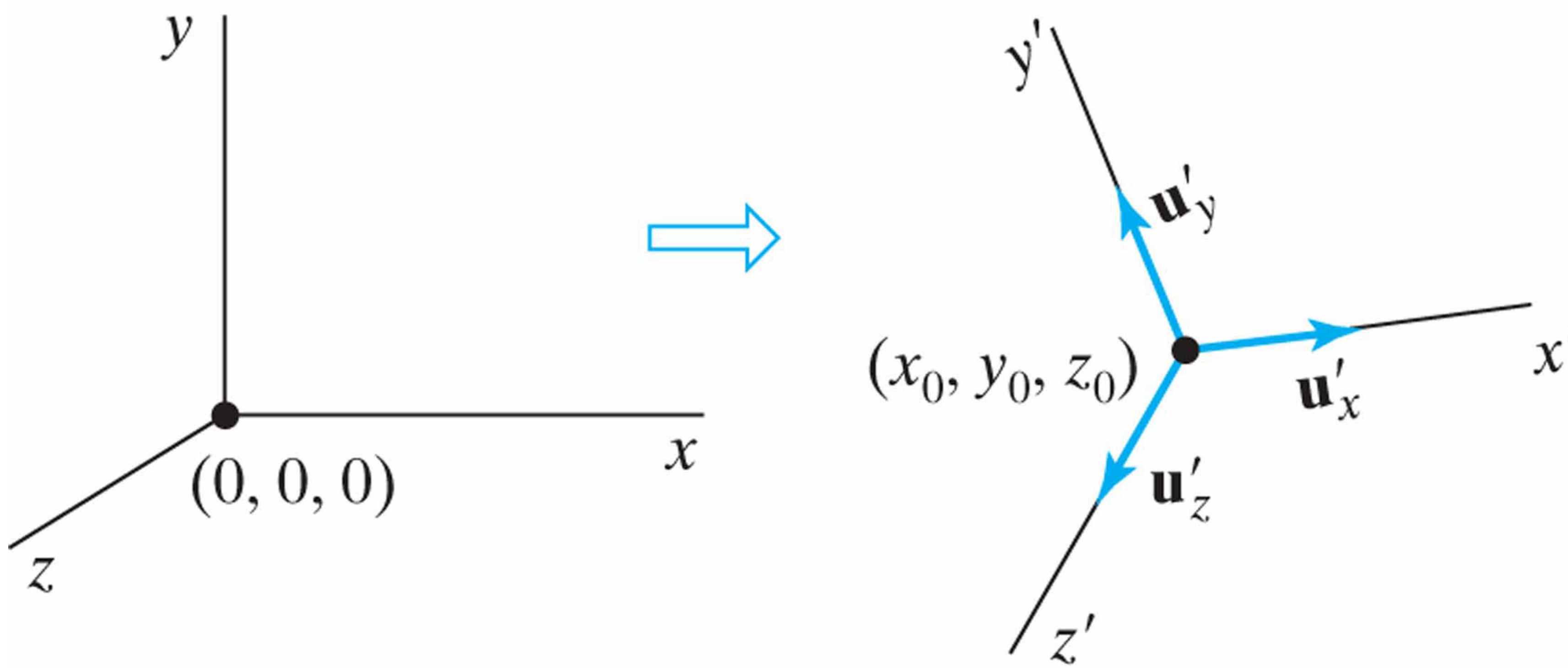
➢ A general z-axis shearing transformation relative to a selected reference position is produced:

$$M_{zshear} = \begin{bmatrix} 1 & 0 & sh_{zx} & -sh_{zx} \cdot z_{ref} \\ 0 & 1 & sh_{zy} & -sh_{zy} \cdot z_{ref} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



7.6 Transformations between Three-Dimensional Coordinate Systems

An x'y'z' coordinate system defined within an x y z system. A scene description is transferred to the new coordinate reference using a transformation sequence that superimposes the x'y'z' frame on the xyz axes.



- Translation

$$\mathbf{T} = (-x_0, -y_0, -z_0)$$

- Rotation

$$\mathbf{R} = \begin{bmatrix} u'_{x1} & u'_{x2} & u'_{x3} & 0 \\ u'_{y1} & u'_{y2} & u'_{y3} & 0 \\ u'_{z1} & u'_{z2} & u'_{z3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

7.7 Affine Transformations

A coordinate transformation of the form

$$x' = a_{xx}x + a_{xy}y + a_{xz}z + b_x$$
$$y' = a_{yx}x + a_{yy}y + a_{yz}z + b_y$$
$$z' = a_{zx}x + a_{zy}y + a_{zz}z + b_z$$

is called **affine transformation**.

- Parallel lines are transformed into parallel lines, and finite points map to finite points.

- Translation, rotation, scaling, reflection, and shear are examples of affine transformations. We can express any affine transformation as some composition of these five transformations.